ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Programming Concepts in Scientific Computing

Installation manual for Sound Processing project

Didier Bieler
Sergio Daniel Hernández Charpak

Professor:
Guillaume Anciaux
Assitants:
Allan S. Nielsen
Luca Pegolotti

December 14, 2017

École polytechnique fédérale de Lausanne
School of Physics & Mathematics
Master semesters 1 and 2
Lausanne, Switzerland

1 Manual Installation and Compilation Manual

The following installation instructions are for a **Linux platform operating system**. They are tested for Ubuntu 14.04 from Canonical. The project uses CMake as compilation helper. If you already have cmake with its interactive option ccmake please skip the CMake section.

1.1 Libraries: CMake

- CMake (version $\geq$ 2.8)

CMake is necessary as it is the control of the software compilation process. This manual implementation illustrates the compilation with the most recent version at the time, 3.10.0. The interactive option of CMake, ccmake is recommended and was used during the current installation. You can get the latest stable version from the Cmake Download Page [1]. A C++ compiler such as *g++* and *make* are necessary. Once the source code is downloaded and extracted, to install cmake, run the *bootstrap* script (here use the *–help* option to see the supported options, such as the *–prefix=custom install directory path* option), then *make* and *sudo make install* (or *make install* if a custom path was selected and no privileges are required). An additional library (*libncurses5-dev*) may be necessary for ccmake.

In summary:

```
1. $ sudo apt-get install libncurses5-dev
2. $ wget https://cmake.org/files/v3.10/cmake-3.10.0.tar.gz
3. $ tar -xvf cmake-3.10.0.tar.gz
4. $ cd cmake-3.10.0/
5. $ ./bootstrap
6. $ make
7. $ sudo make install
```

1.2 Sound Processing Project

- Getting and Compiling the Project

Clone the public repository using git. Then create a new folder (where the code will be built). Move to the new folder folder and run ccmake from the build folder with argument the source folder of the project (located in the subfolder Project of the repository). Press *c* to configure and, as there are no special libraries to be imported

[1] https://cmake.org/cmake/resources/software.html

(if cmake is properly installed) you should get the following on your terminal:

CMAKE_BUILD_TYPE :
CMAKE_INSTALL_PREFIX : /usr/local

Then configure again by pressing *c* and generate the Makefile by pressing *g*. Then build the code by running *make*.

In summary:

```
8.  $ git clone https://github.com/sercharpak/PCSC2017_Group4.git
9.  $ mkdir Project_Built
10. $ cd Project_Built
11. $ ccmake ../PCSC2017_Group4/Project/
12. Press c
13. Make sure the following options have the correct values:
    CMAKE_BUILD_TYPE :
    CMAKE_INSTALL_PREFIX : /usr/local
14. Press c again
15. Press g
16. $ make
```

If all these steps were correctly followed and no errors were obtained, the program is ready to be used.

- Building the Doxygen Documentation (optional)

The documentation is already provided. Nevertheless, if you desire to regenerate it following modifications of the source code it is possible. For this **Doygen** is a dependency. You can install it following the instructions on the Doxygen website[2]. Once it is installed, just go to the project built folder and run *make doc*.

In summary:

```
17.  Install Doxygen
18. $ cd Project_Built
19. $ make doc
```

[2] www.doxygen.org/